Stable Project Matching in R

IEOR 115 DP

The Berkeley Group

We can use three packages in `R` to find a stable project matching based on the preferences of the consultants and projects leads. These packages are `DataComputing`, `matchingR` and `readxl`.

STEP 1 Install the packages if you don't already have them

```
install.packages("DataComputing")
install.packages("matchingR")
install.packages("readxl")
```

STEP 2 Load the packages

```
library(DataComputing)
library(matchingR)
library(readxl)
```

STEP 3 Load the preference tables you exported from Access and convert Rank to be numeric

```
consultant_pref <- read_excel("Project_Rank Query.xlsx")#change path directory accordingly
consultant_pref$Rank <- as.numeric(consultant_pref$Rank)
plead_pref <- read_excel("Consultant_Rank Query.xlsx")#change path directory accordingly
plead_pref$Rank <- as.numeric(plead_pref$Rank)
plead_pref
```

```
## Source: local data frame [12 x 3]
##
##    Project_lead Consultant  Rank
##           (chr)      (chr) (dbl)
## 1          Mary     Justin     1
## 2          Mary  Achilleas     2
## 3          Mary     Nicole     5
## 4          Mary       Matt     6
## 5          Mary     Taylor     4
## 6          Mary       Joao     3
## 7         Tejas     Justin     1
## 8         Tejas  Achilleas     5
## 9         Tejas     Nicole     2
## 10        Tejas       Matt     3
## 11        Tejas     Taylor     4
## 12        Tejas       Joao     6
```

```
consultant_pref
```

```
## Source: local data frame [12 x 3]
##
##    Consultant Project_lead  Rank
```

```
##         (chr)        (chr) (dbl)
## 1      Justin         Mary     1
## 2      Justin        Tejas     2
## 3        Matt         Mary     1
## 4        Matt        Tejas     2
## 5      Taylor         Mary     2
## 6      Taylor        Tejas     1
## 7      Nicole         Mary     1
## 8      Nicole        Tejas     2
## 9        Joao         Mary     2
## 10       Joao        Tejas     1
## 11  Achilleas         Mary     2
## 12  Achilleas        Tejas     1
```

STEP 4 Wrangle the data in the following way to produce the following preference tables and create two tables `consultants` and `project_leads` which will contain the newly assigned ID and name for each group.

```
pref_plead <- spread(plead_pref, Project_lead, Rank)
pref_plead <- pref_plead[order(pref_plead$Consultant),]

pref_cons <- spread(consultant_pref, Consultant, Rank)
pref_cons <- pref_cons[order(pref_cons$Project_lead),]

project_leads <- cbind(unique(row.names(pref_cons)),pref_cons[,1])
colnames(project_leads)  <- c("projectlead_id","name")
project_leads$projectlead_id <- as.numeric(project_leads$projectlead_id)
consultants <-  cbind(unique(row.names(pref_plead)),pref_plead[,1])
colnames(consultants)  <- c("consultant_id","name")
consultants$consultant_id <- as.numeric(consultants$consultant_id)

pref_cons <- pref_cons[,-1]
pref_plead <- pref_plead[,-1]
pref_plead
```

```
## Source: local data frame [6 x 2]
##
##    Mary Tejas
##   (dbl) (dbl)
## 1     2     5
## 2     3     6
## 3     1     1
## 4     6     3
## 5     5     2
## 6     4     4
```

```
pref_cons
```

```
## Source: local data frame [2 x 6]
##
##   Achilleas  Joao Justin  Matt Nicole Taylor
##       (dbl) (dbl)  (dbl) (dbl)  (dbl)  (dbl)
## 1         2     2      1     1      1      2
## 2         1     1      2     2      2      1
```

```
consultants
```

```
##   consultant_id      name
## 1             1 Achilleas
## 2             2      Joao
## 3             3    Justin
## 4             4      Matt
## 5             5    Nicole
## 6             6    Taylor
```

```
project_leads
```

```
##   projectlead_id  name
## 1              1  Mary
## 2              2 Tejas
```

STEP 5 Use the `galeShapley.collegeAdmissions` function in the `matchingR` package to create a stable match.

```
n <-  3 # specify how many people can be assigned to each group
stable_match <- galeShapley.collegeAdmissions(pref_cons,pref_plead,slots = n,
                                              studentOptimal = TRUE)
```

STEP 6 Using the output and the tables created earlier we can play with the data to make the results more readable

```
consultant_matches <- cbind(consultants,stable_match$matched.students)
consultant_matches <- left_join(consultant_matches,project_leads,
                                by = c("stable_match$matched.students" = "projectlead_id"))

consultant_matches_final <-
  consultant_matches %>%
  select(name.x,name.y) %>%
  rename(Consultant = name.x, Project_Lead = name.y)

project_lead_matches <- cbind(project_leads,stable_match$matched.colleges)
project_lead_matches <- project_lead_matches[,-1]
project_lead_matches <- gather(project_lead_matches, name, consultant_id)
project_lead_matches <-  project_lead_matches[,-2]
colnames(project_lead_matches) <- c("Project_Lead", "consultant_id")


project_lead_matches_final <-
  project_lead_matches %>%
  left_join(consultants) %>%
  select(-consultant_id) %>%
  rename(Consultant=name) %>%
  arrange(Project_Lead)
```

```
## Joining by: "consultant_id"
```

STEP 7 Print the final tables containing the stable project pairings

```
consultant_matches_final
```

```
##   Consultant Project_Lead
## 1  Achilleas         Mary
## 2       Joao         Mary
## 3     Justin        Tejas
## 4       Matt        Tejas
## 5     Nicole        Tejas
## 6     Taylor         Mary
```

```
project_lead_matches_final
```

```
##   Project_Lead Consultant
## 1         Mary     Taylor
## 2         Mary       Joao
## 3         Mary  Achilleas
## 4        Tejas       Matt
## 5        Tejas     Nicole
## 6        Tejas     Justin
```